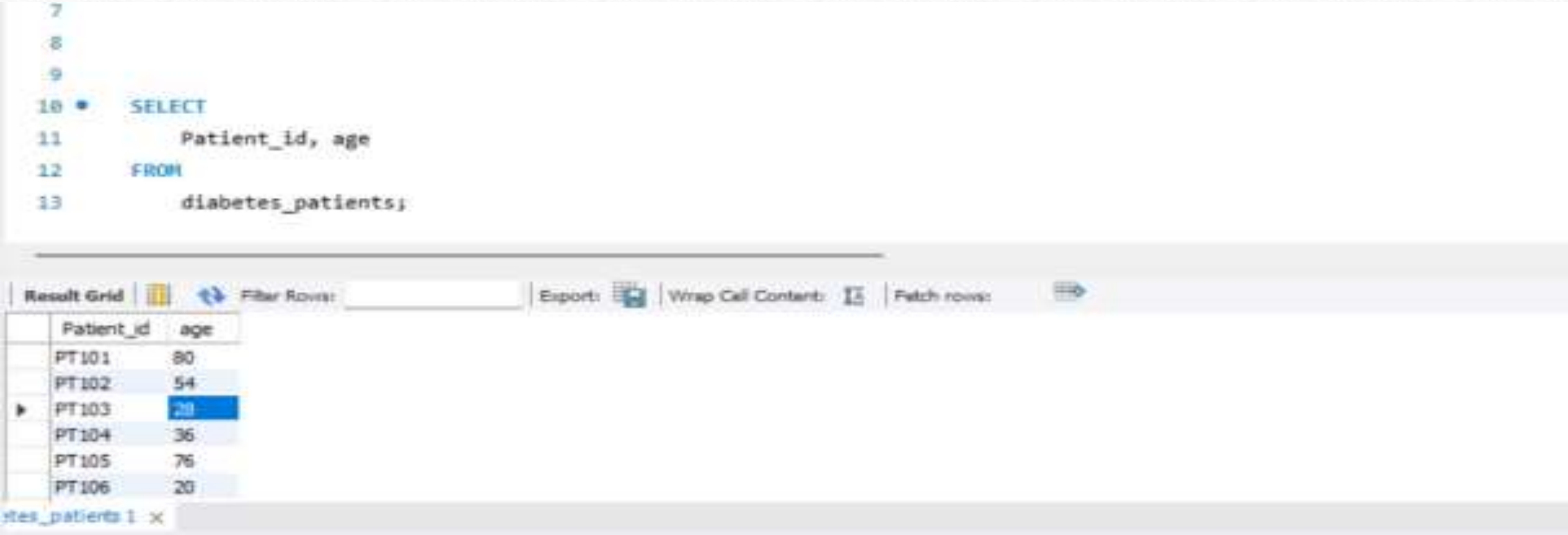
DIABETES PREDICATION ASSESSMENT

TANYA MITTAL (DATA ANALYST INTERN)

INTRODUCTION

The provided dataset by psyliq comprises detailed information about diabetes patients, which includes patient ID, age, gender, body mass index, blood pressure and few more columns related to it. This dataset comprises approximately 100,000 rows and 11 columns. To extract relevant insights from the dataset, the analysis has been performed using the MYSQL tool, which focuses on extracting meaningful insights. This tool provides a comprehensive and flexible environment for working with large datasets, enabling analysts to query and manipulate the data in various ways.

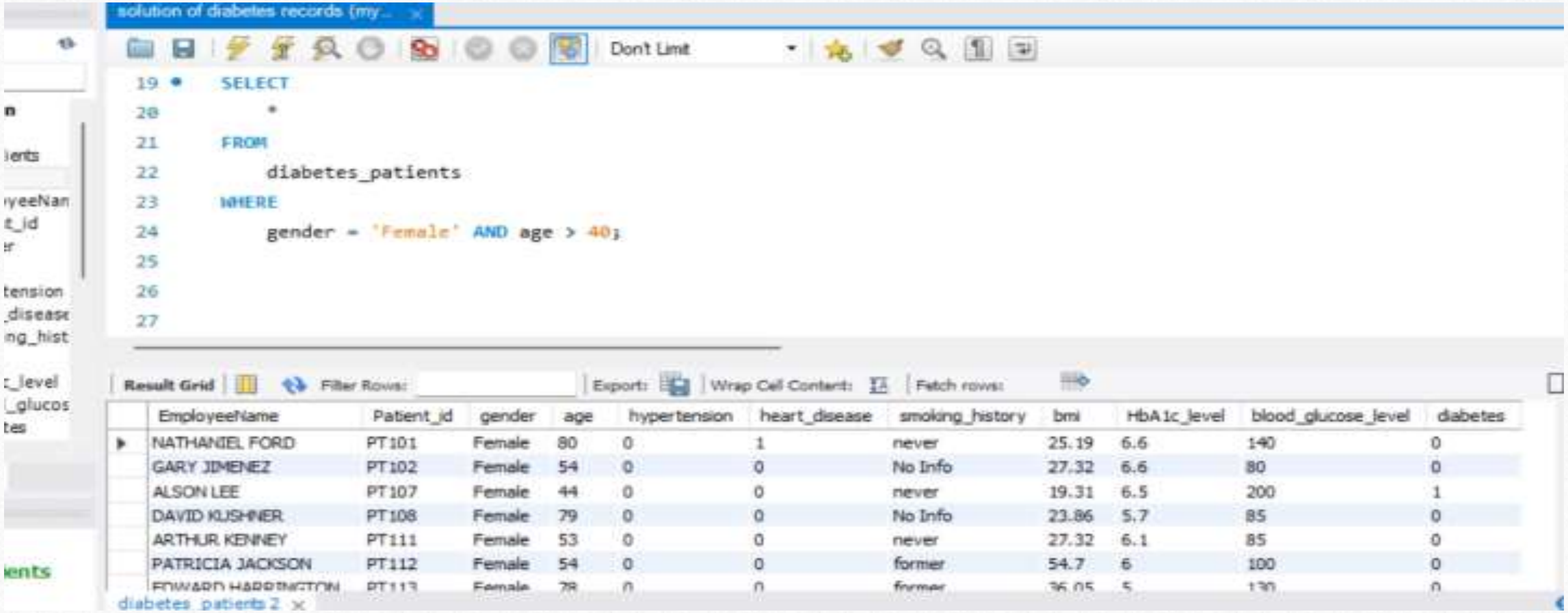
Retrieve the Patient_id and ages of all patients.



```
SELECT
    Patient_id, age
FROM
    diabetes_patients;
```

Patient_id	age
PT101	80
PT102	54
PT103	28
PT104	36
PT105	76
PT106	20

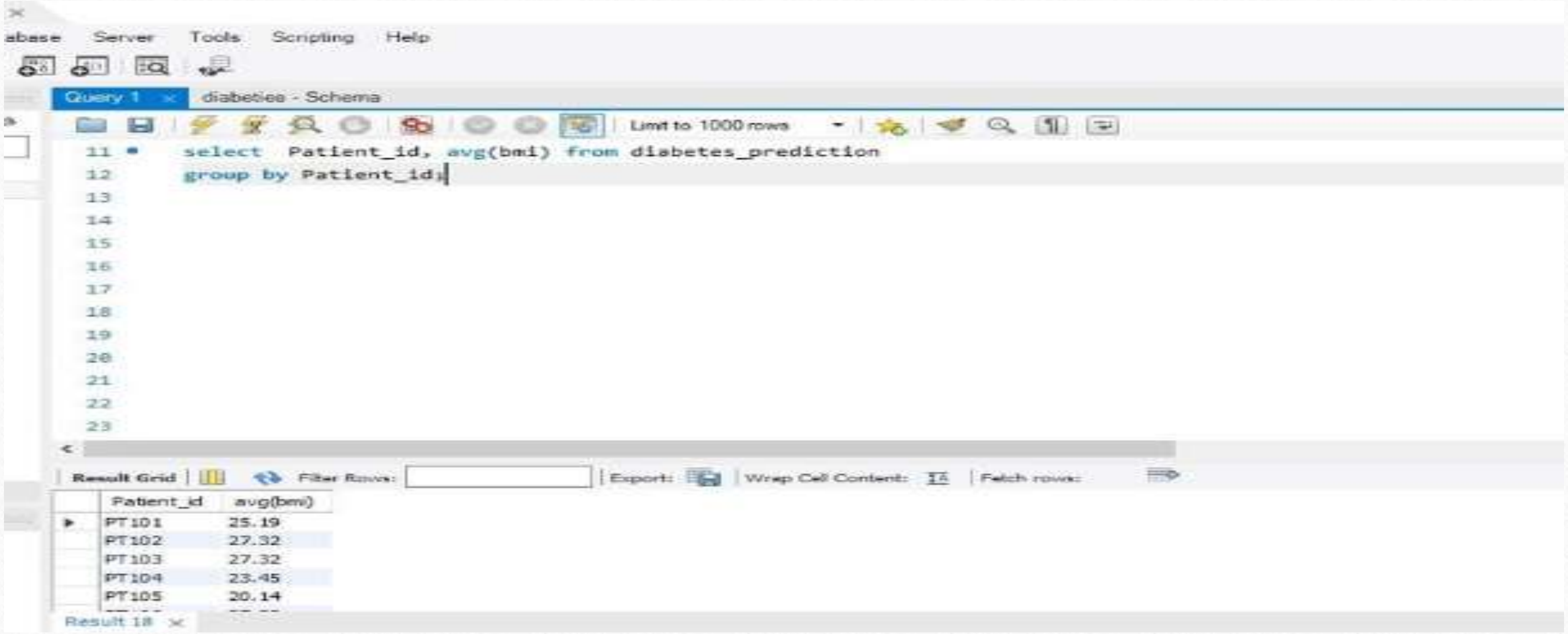
Select all female patients who are older than 40.



```sql
SELECT
    *
FROM
    diabetes_patients
WHERE
    gender = 'Female' AND age > 40;
```

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
NATHANIEL FORD	PT101	Female	80	0	1	never	25.19	6.6	140	0
GARY JIMENEZ	PT102	Female	54	0	0	No Info	27.32	6.6	80	0
ALSON LEE	PT107	Female	44	0	0	never	19.31	6.5	200	1
DAVID KUSHNER	PT108	Female	79	0	0	No Info	23.86	5.7	85	0
ARTHUR KENNEY	PT111	Female	53	0	0	never	27.32	6.1	85	0
PATRICIA JACKSON	PT112	Female	54	0	0	former	54.7	6	100	0
EDWARD HARRINGTON	PT113	Female	78	0	0	former	36.05	5	130	0

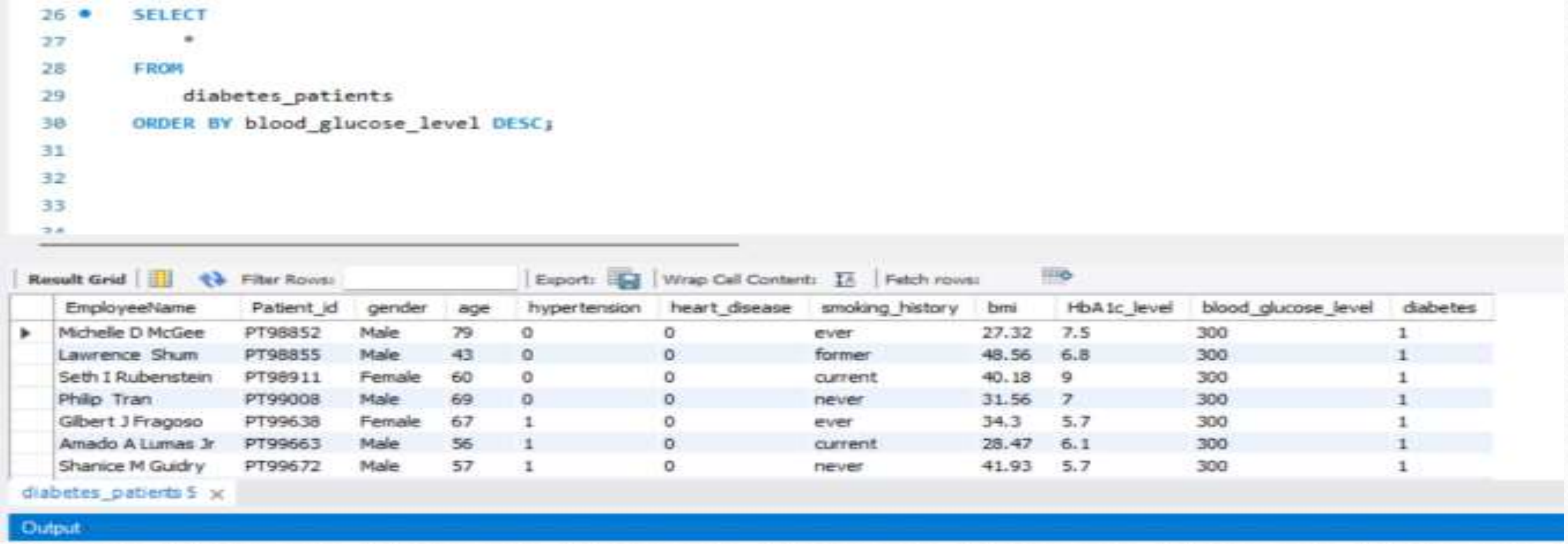
Calculate the average BMI of patients.



```sql
select  Patient_id, avg(bmi) from diabetes_prediction
group by Patient_id;
```

Patient_id	avg(bmi)
PT101	25.19
PT102	27.32
PT103	27.32
PT104	23.45
PT105	20.14

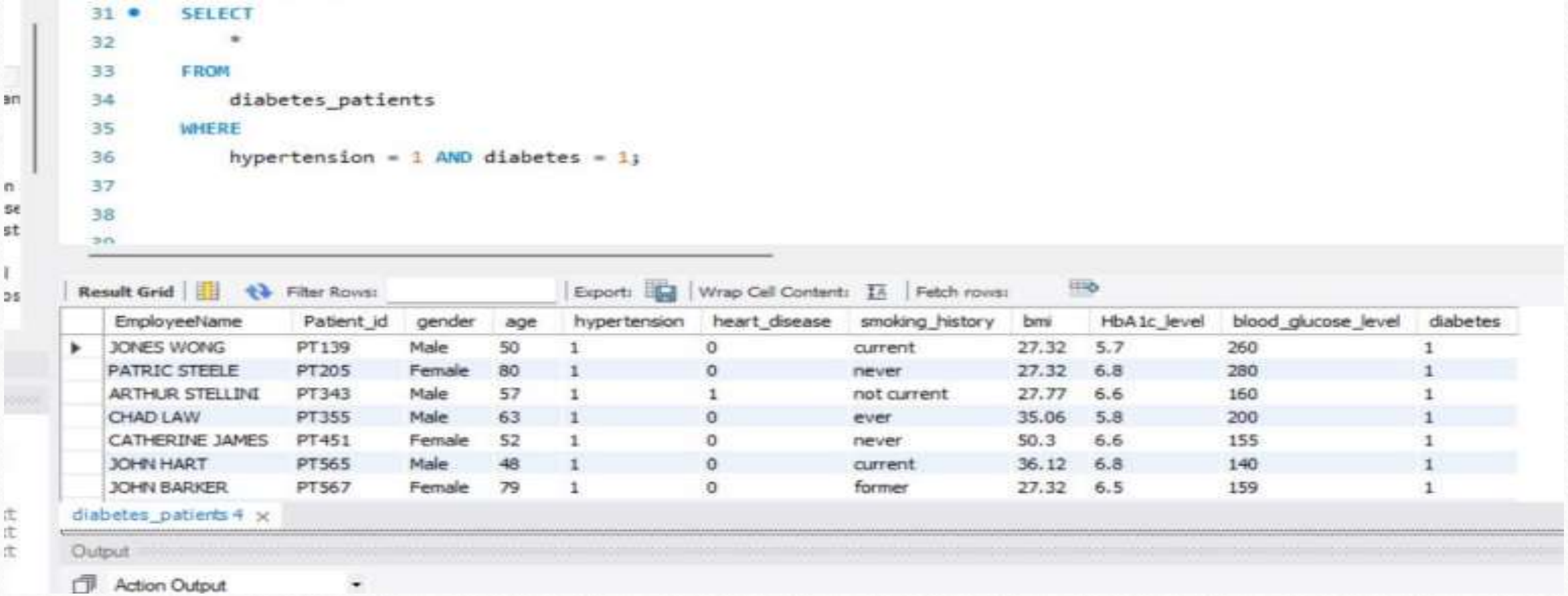
List patients in descending order of blood glucose levels.



```
SELECT
    *
FROM
    diabetes_patients
ORDER BY blood_glucose_level DESC;
```

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Michelle D McGee	PT98852	Male	79	0	0	ever	27.32	7.5	300	1
Lawrence Shum	PT98855	Male	43	0	0	former	48.56	6.8	300	1
Seth I Rubenstein	PT98911	Female	60	0	0	current	40.18	9	300	1
Philip Tran	PT99008	Male	69	0	0	never	31.56	7	300	1
Gilbert J Fragoso	PT99638	Female	67	1	0	ever	34.3	5.7	300	1
Amado A Lumas Jr	PT99663	Male	56	1	0	current	28.47	6.1	300	1
Shanice M Guidry	PT99672	Male	57	1	0	never	41.93	5.7	300	1

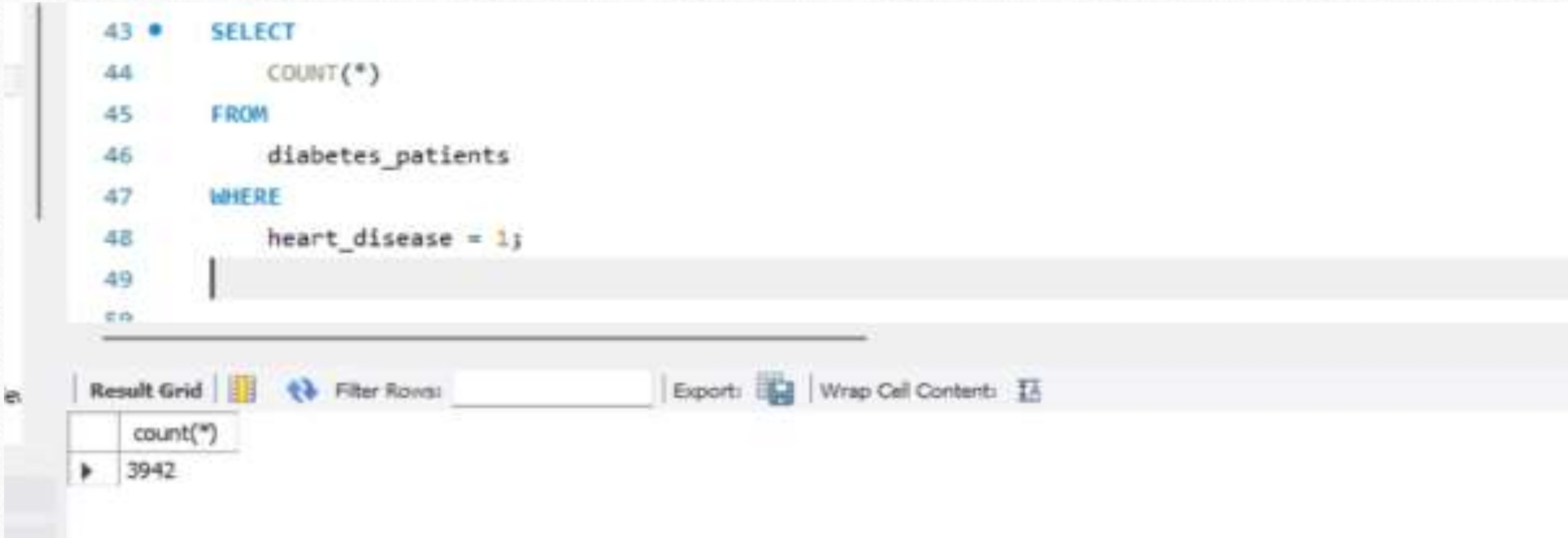
Find patients who have hypertension and diabetes.



```
SELECT
    *
FROM
    diabetes_patients
WHERE
    hypertension = 1 AND diabetes = 1;
```

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
▶	JONES WONG	PT139	Male	50	1	0	current	27.32	5.7	260	1
	PATRIC STEELE	PT205	Female	80	1	0	never	27.32	6.8	280	1
	ARTHUR STELLINI	PT343	Male	57	1	1	not current	27.77	6.6	160	1
	CHAD LAW	PT355	Male	63	1	0	ever	35.06	5.8	200	1
	CATHERINE JAMES	PT451	Female	52	1	0	never	50.3	6.6	155	1
	JOHN HART	PT565	Male	48	1	0	current	36.12	6.8	140	1
	JOHN BARKER	PT567	Female	79	1	0	former	27.32	6.5	159	1

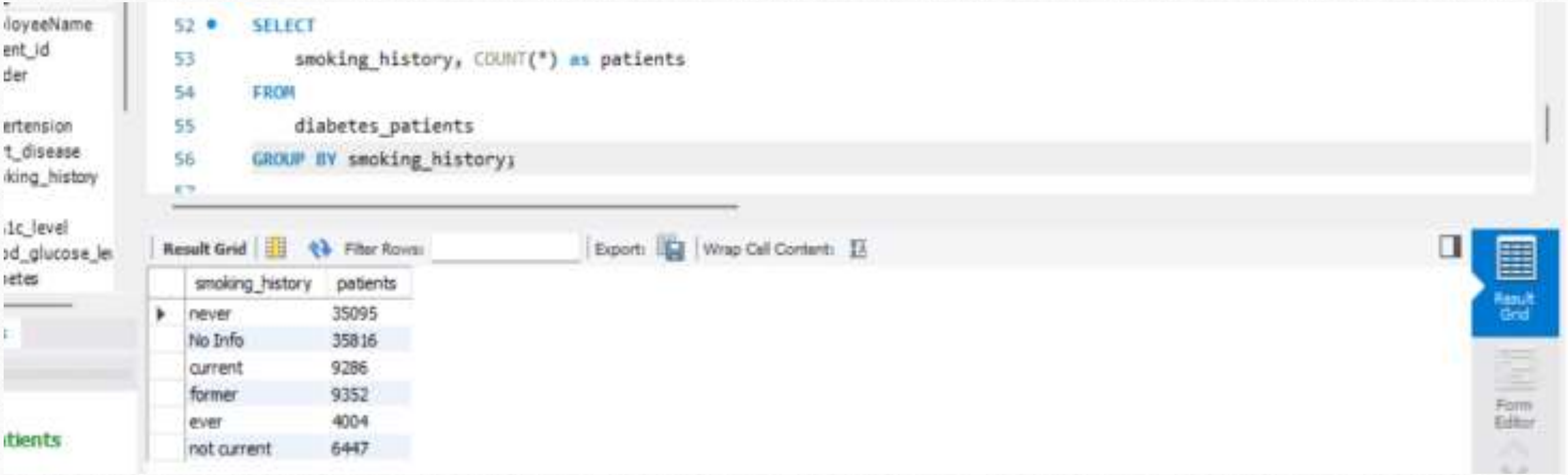
Determine the number of patients with heart disease.



```
SELECT
    COUNT(*)
FROM
    diabetes_patients
WHERE
    heart_disease = 1;
```

```
count(*)
3942
```

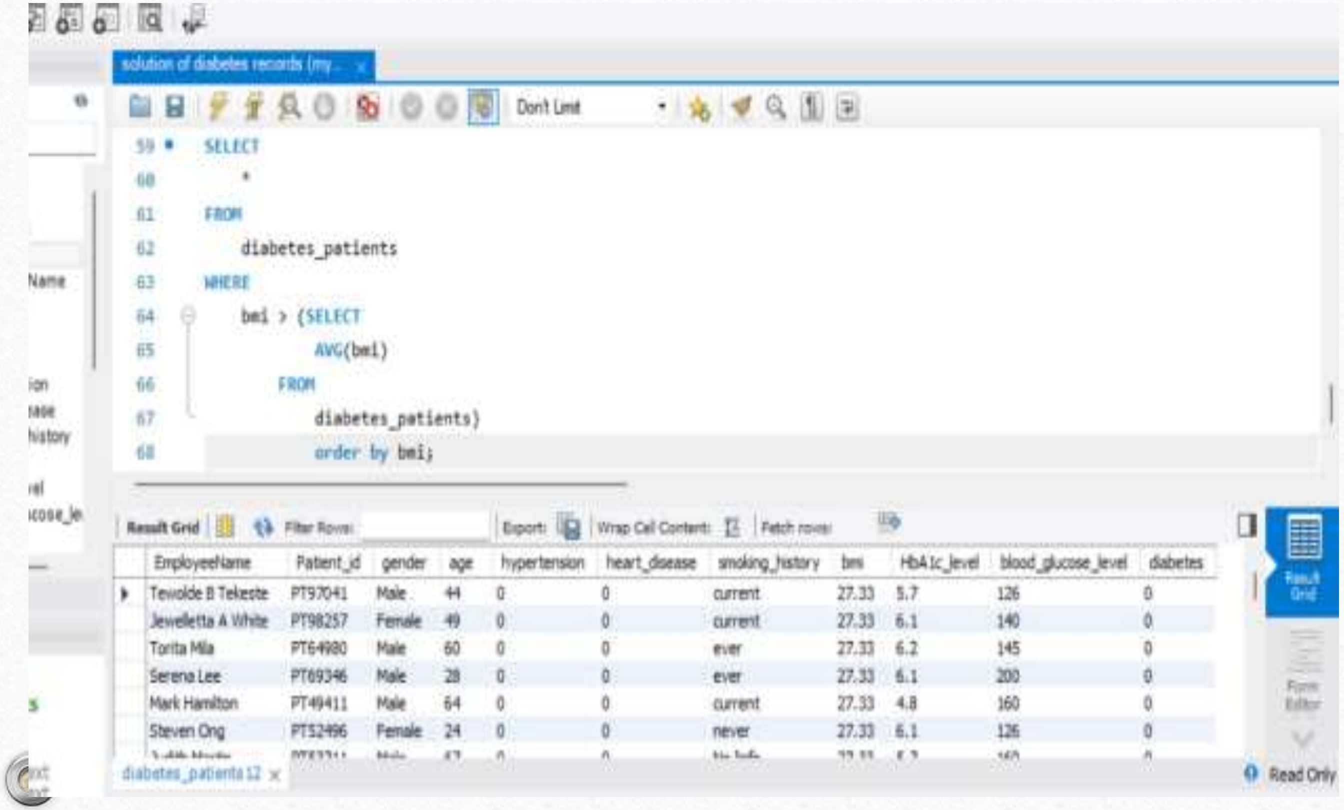
Group patients by smoking history and count how many smokers and non-smokers there are.



```
SELECT
    smoking_history, COUNT(*) as patients
FROM
    diabetes_patients
GROUP BY smoking_history;
```

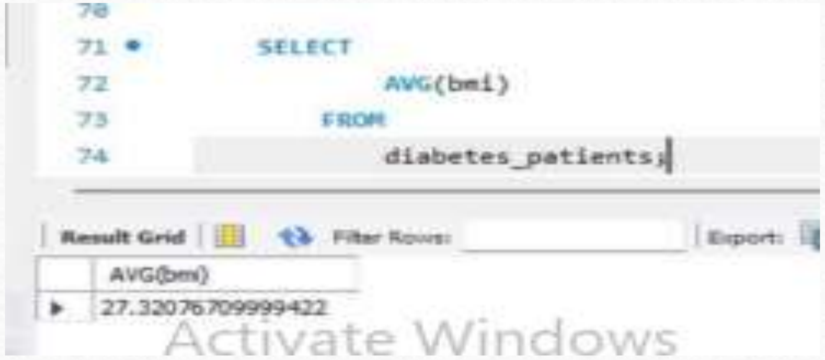
smoking_history	patients
never	35095
No Info	35816
current	9286
former	9352
ever	4004
not current	6447

Retrieve the Patient_ids of patients who have a BMI greater than the average BMI.



```
SELECT
    *
FROM
    diabetes_patients
WHERE
    bmi > (SELECT
            AVG(bmi)
        FROM
            diabetes_patients)
            order by bmi;
```

	EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
▸	Tewolde B Tekeste	PT97041	Male	44	0	0	current	27.33	5.7	126	0
	Jewelletta A White	PT98257	Female	49	0	0	current	27.33	6.1	140	0
	Torita Mila	PT64980	Male	60	0	0	ever	27.33	6.2	145	0
	Serena Lee	PT69346	Male	28	0	0	ever	27.33	6.1	200	0
	Mark Hamilton	PT49411	Male	64	0	0	current	27.33	4.8	160	0
	Steven Ong	PT52496	Female	24	0	0	never	27.33	6.1	126	0



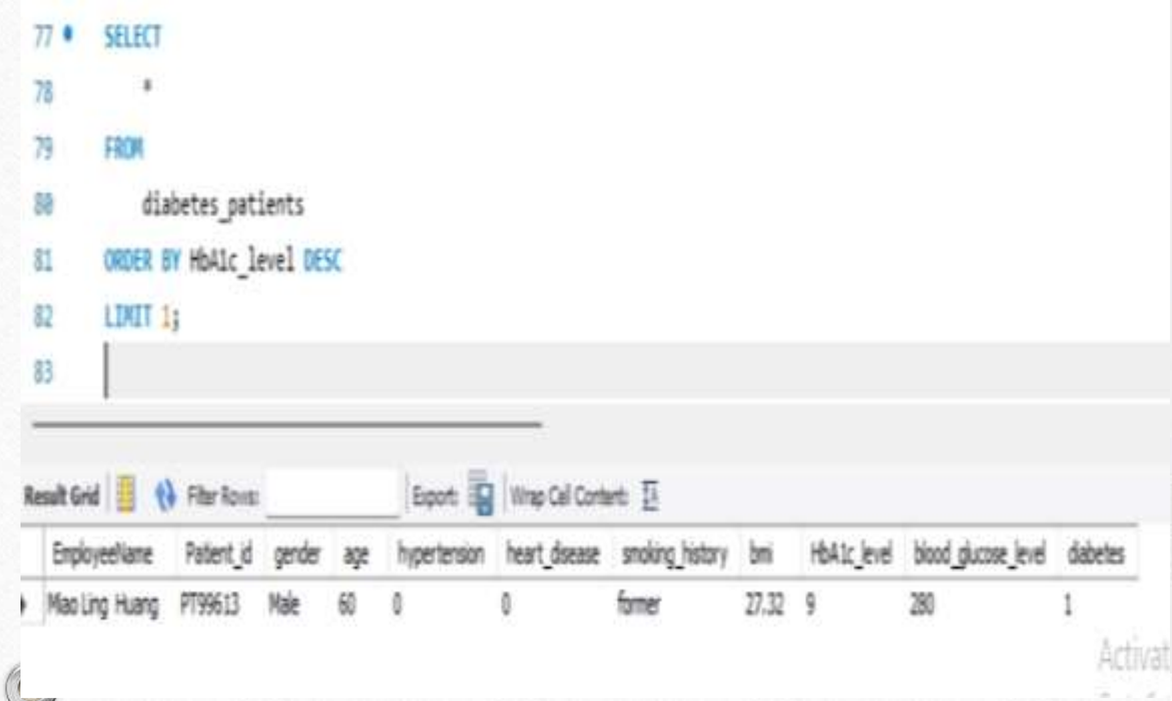
```
SELECT
    AVG(bmi)
FROM
    diabetes_patients;
```

	AVG(bmi)
▸	27.320767099999422

Activate Windows

Find the patient with the highest HbA1c level and the patient with the lowest HbA1clevel.

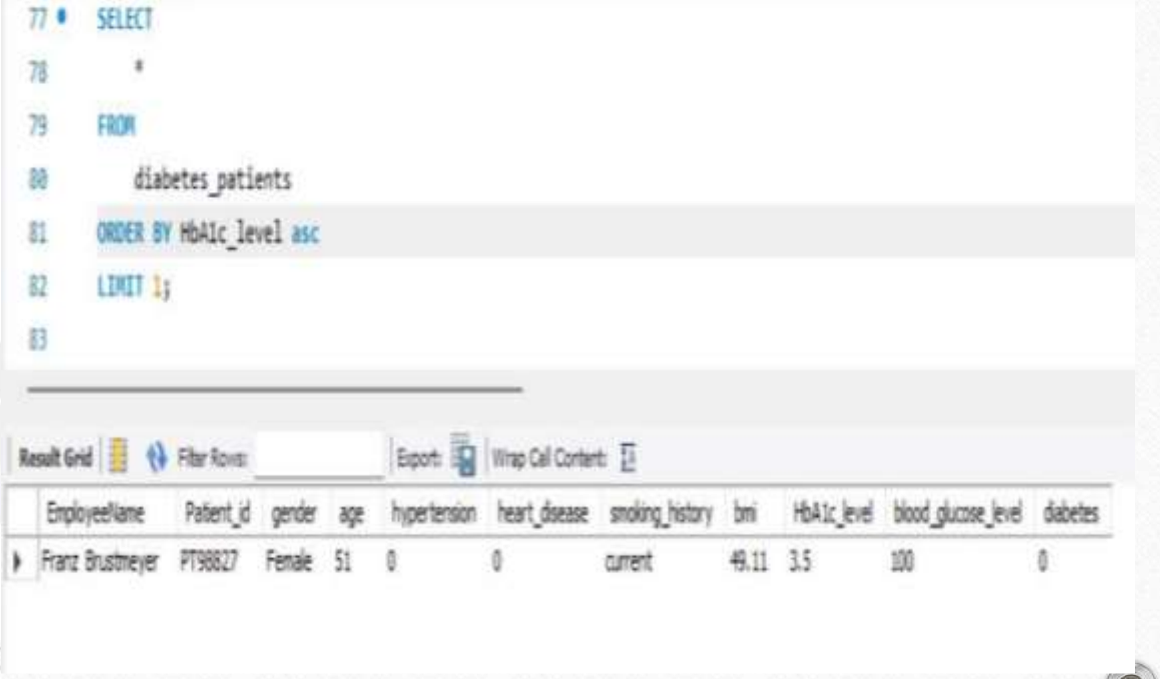
highest HbA1c level



```
SELECT
    *
FROM
    diabetes_patients
ORDER BY HbA1c_level DESC
LIMIT 1;
```

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Mao Ling Huang	PT99613	Male	60	0	0	former	27.32	9	280	1

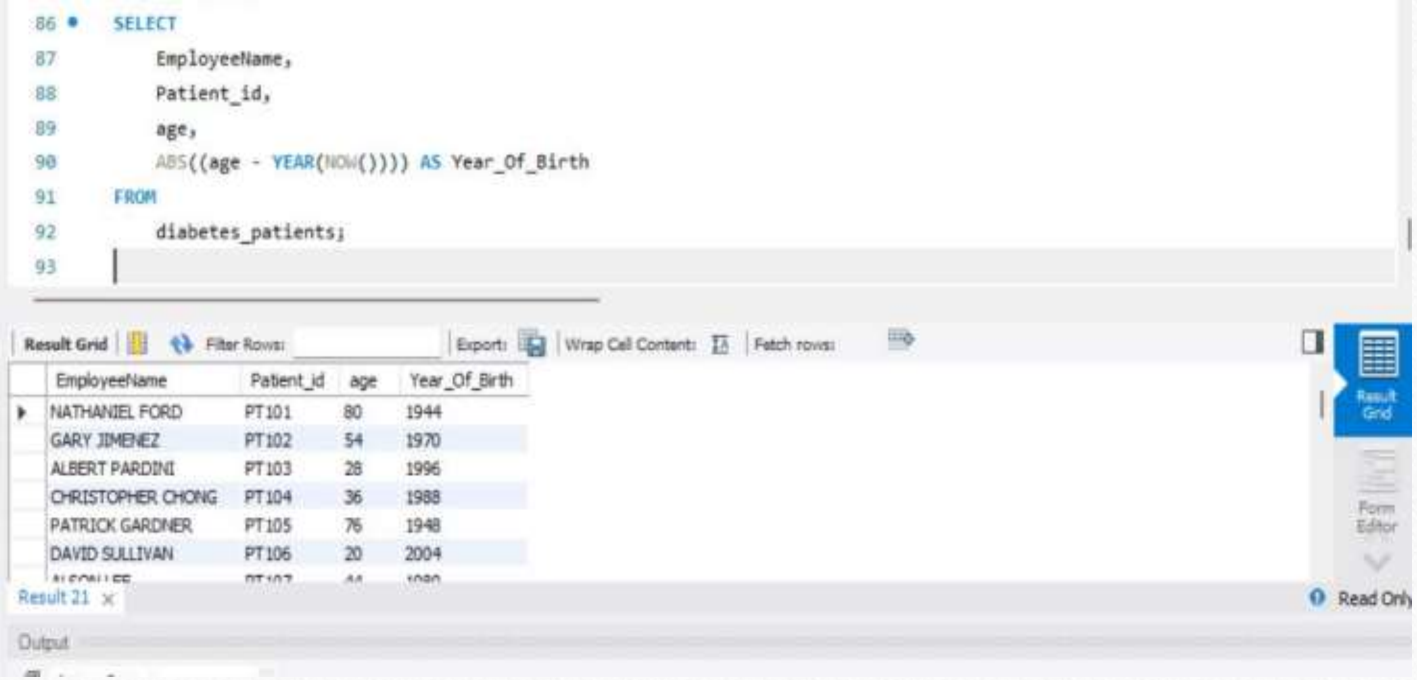
Lowest HbA1c level



```
SELECT
    *
FROM
    diabetes_patients
ORDER BY HbA1c_level asc
LIMIT 1;
```

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Franz Brustmeyer	PT98827	Female	51	0	0	current	49.11	3.5	100	0

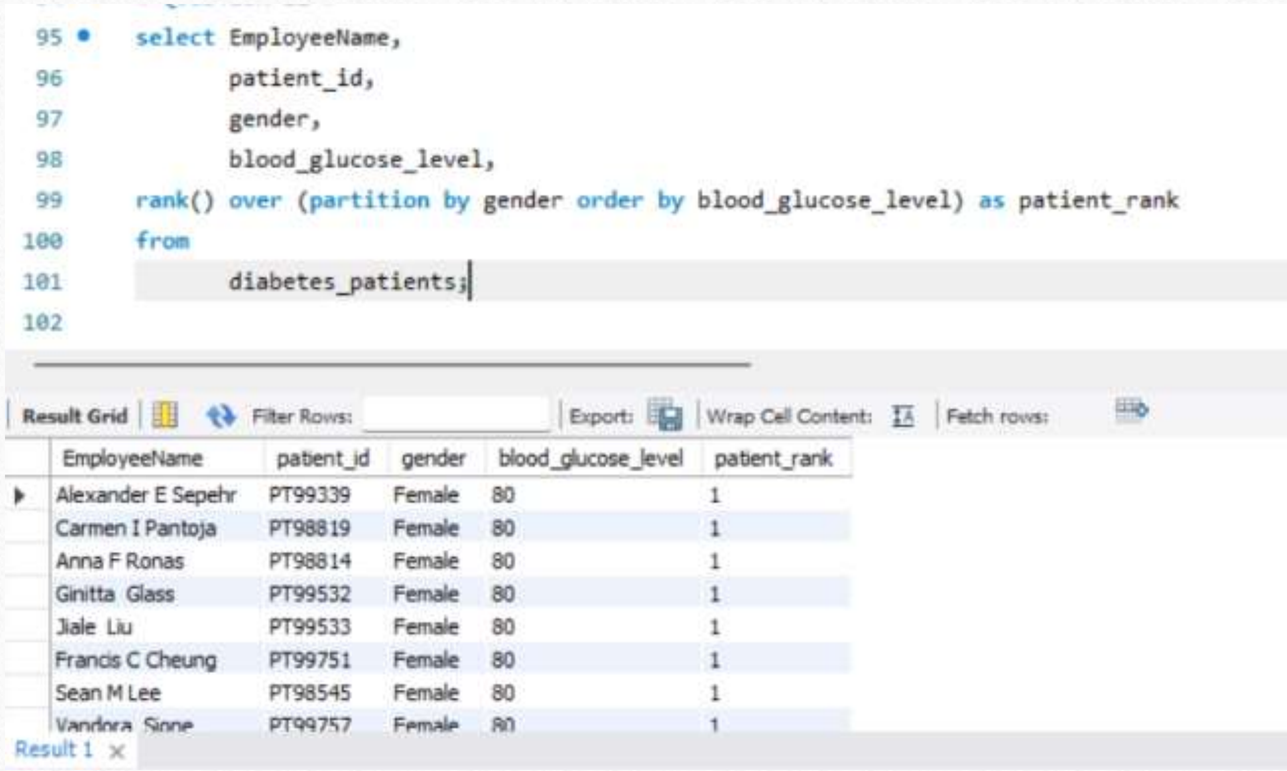
Calculate the age of patients in years (assuming the current date as of now).



```sql
SELECT
    EmployeeName,
    Patient_id,
    age,
    ABS((age - YEAR(NOW()))) AS Year_Of_Birth
FROM
    diabetes_patients;
```

EmployeeName	Patient_id	age	Year_Of_Birth
NATHANIEL FORD	PT101	80	1944
GARY JIMENEZ	PT102	54	1970
ALBERT PARDINI	PT103	28	1996
CHRISTOPHER CHONG	PT104	36	1988
PATRICK GARDNER	PT105	76	1948
DAVID SULLIVAN	PT106	20	2004
ALISON LEE	PT107	44	1980

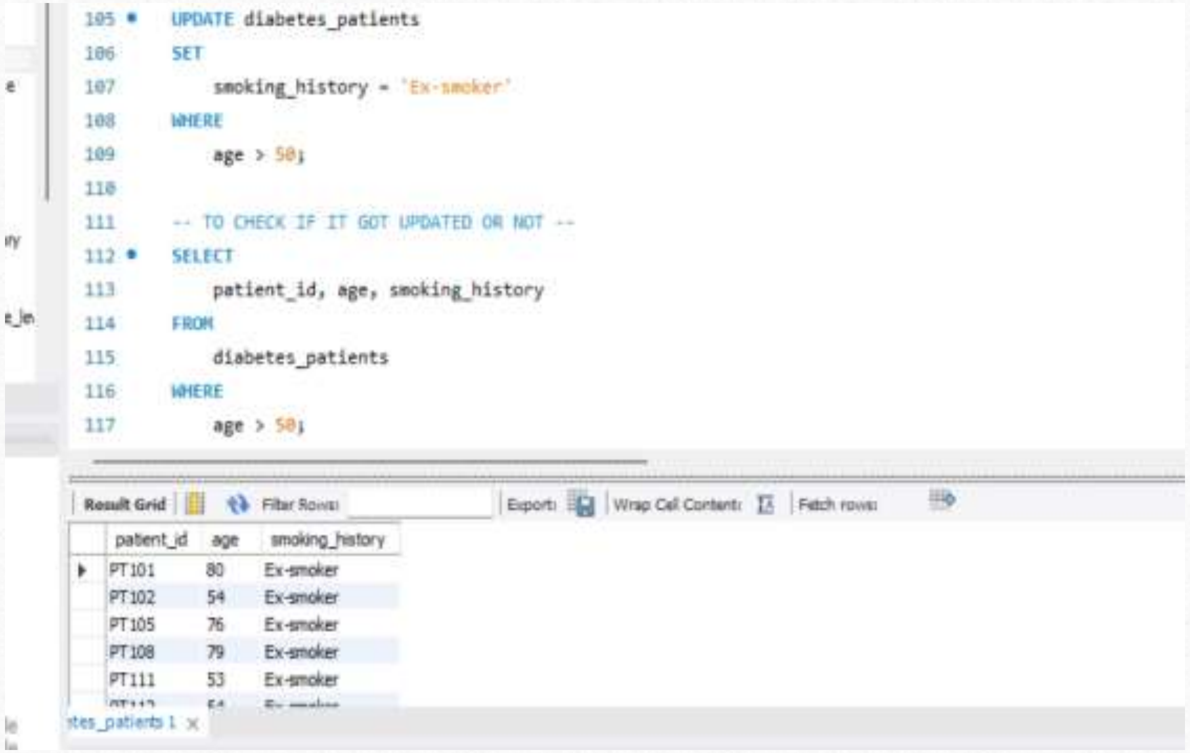
Rank patients by blood glucose level within each gender group.



```
select EmployeeName,
       patient_id,
       gender,
       blood_glucose_level,
rank() over (partition by gender order by blood_glucose_level) as patient_rank
from
       diabetes_patients;
```

EmployeeName	patient_id	gender	blood_glucose_level	patient_rank
Alexander E Sepehr	PT99339	Female	80	1
Carmen I Pantoja	PT98819	Female	80	1
Anna F Ronas	PT98814	Female	80	1
Ginitta Glass	PT99532	Female	80	1
Jiale Liu	PT99533	Female	80	1
Francis C Cheung	PT99751	Female	80	1
Sean M Lee	PT98545	Female	80	1
Vandora Sione	PT99757	Female	80	1

Update the smoking history of patients who are older than 50 to "Ex-smoker."

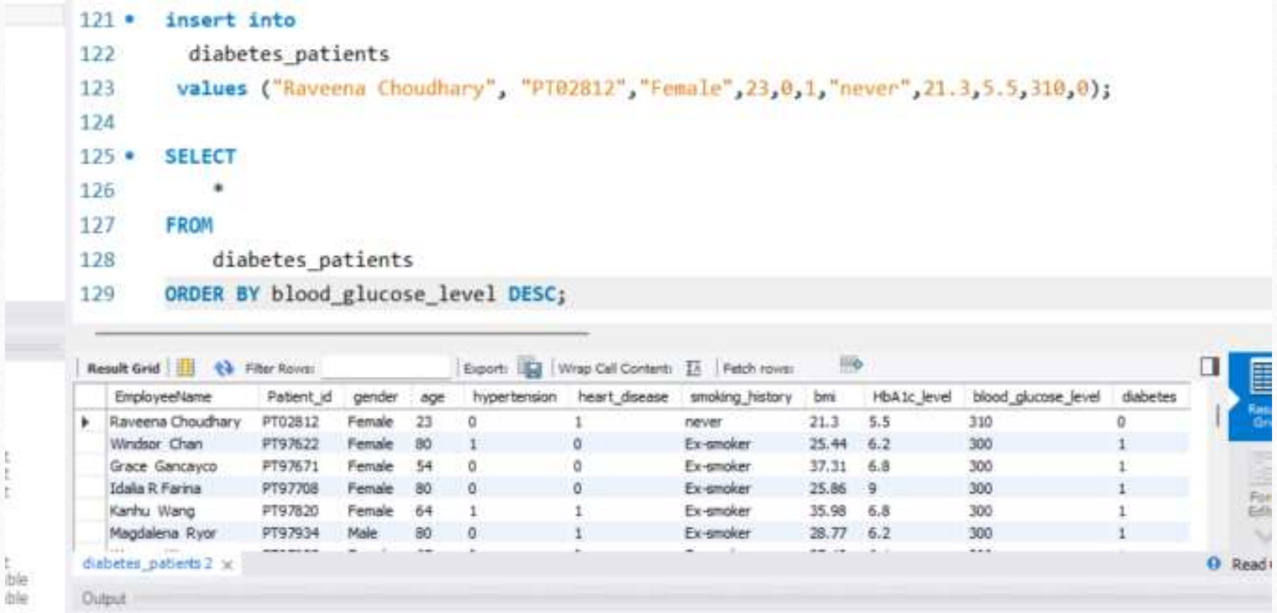


```
UPDATE diabetes_patients
SET
    smoking_history = 'Ex-smoker'
WHERE
    age > 50;

-- TO CHECK IF IT GOT UPDATED OR NOT --
SELECT
    patient_id, age, smoking_history
FROM
    diabetes_patients
WHERE
    age > 50;
```

```
patient_id  age  smoking_history
PT101       80   Ex-smoker
PT102       54   Ex-smoker
PT105       76   Ex-smoker
PT108       79   Ex-smoker
PT111       53   Ex-smoker
PT112       54   Ex-smoker
```

Insert a new patient into the database with sample data.

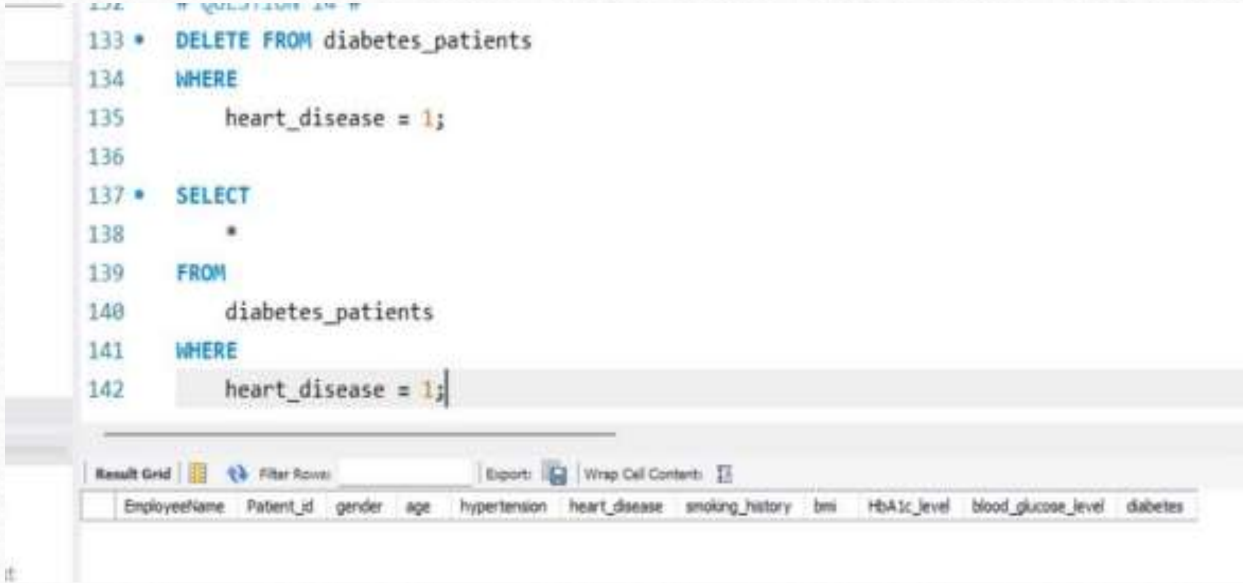


```
insert into
  diabetes_patients
 values ("Raveena Choudhary", "PT02812","Female",23,0,1,"never",21.3,5.5,310,0);

SELECT
    *
FROM
    diabetes_patients
ORDER BY blood_glucose_level DESC;
```

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Raveena Choudhary	PT02812	Female	23	0	1	never	21.3	5.5	310	0
Windsor Chan	PT97622	Female	80	1	0	Ex-smoker	25.44	6.2	300	1
Grace Gancayco	PT97671	Female	54	0	0	Ex-smoker	37.31	6.8	300	1
Idalia R Farina	PT97708	Female	80	0	0	Ex-smoker	25.86	9	300	1
Kanhu Wang	PT97820	Female	64	1	1	Ex-smoker	35.98	6.8	300	1
Magdalena Ryor	PT97934	Male	80	0	1	Ex-smoker	28.77	6.2	300	1

Delete all patients with heart disease from the database.

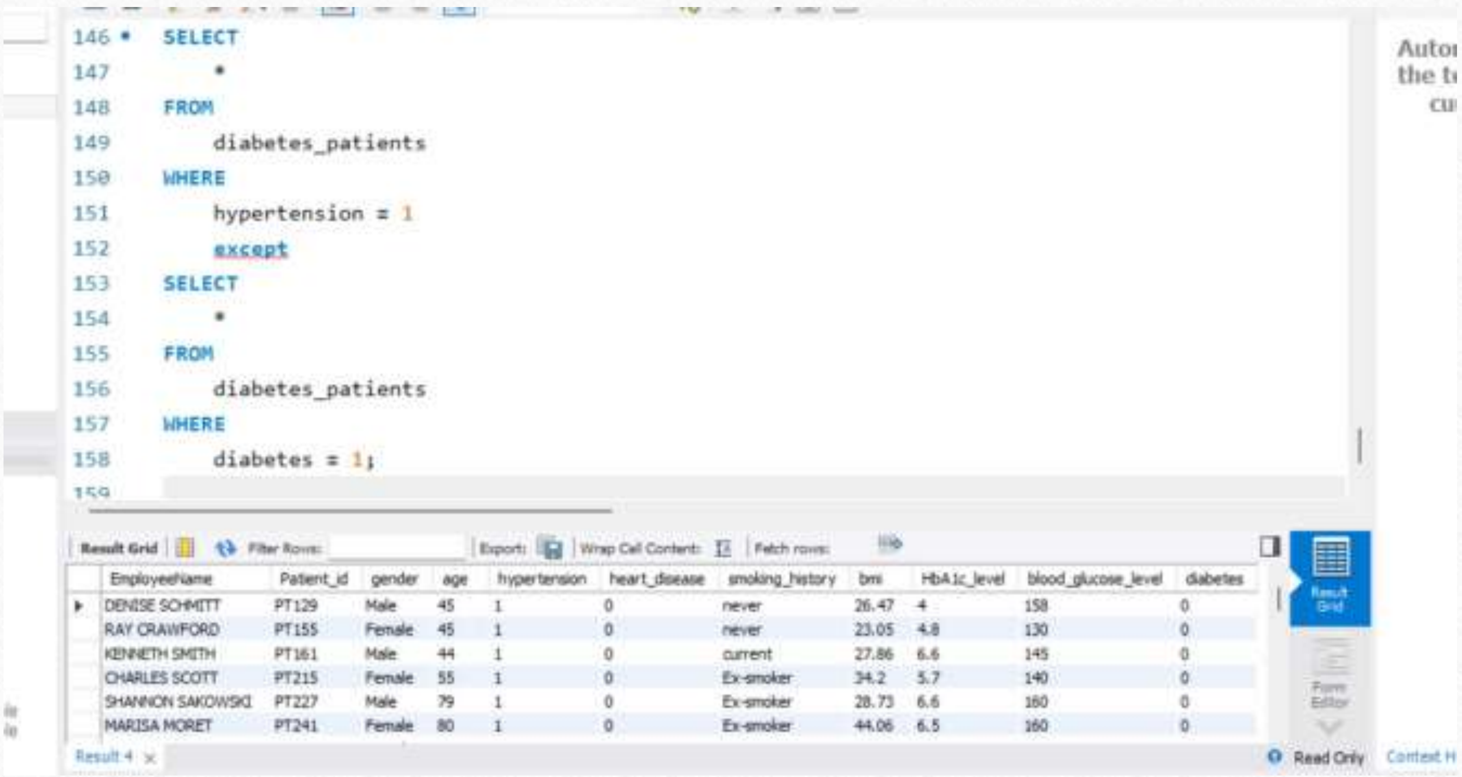


```sql
# QUESTION 14 #
DELETE FROM diabetes_patients
WHERE
    heart_disease = 1;

SELECT
    *
FROM
    diabetes_patients
WHERE
    heart_disease = 1;
```

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes

Find patients who have hypertension but not diabetes using the EXCEPT operator.

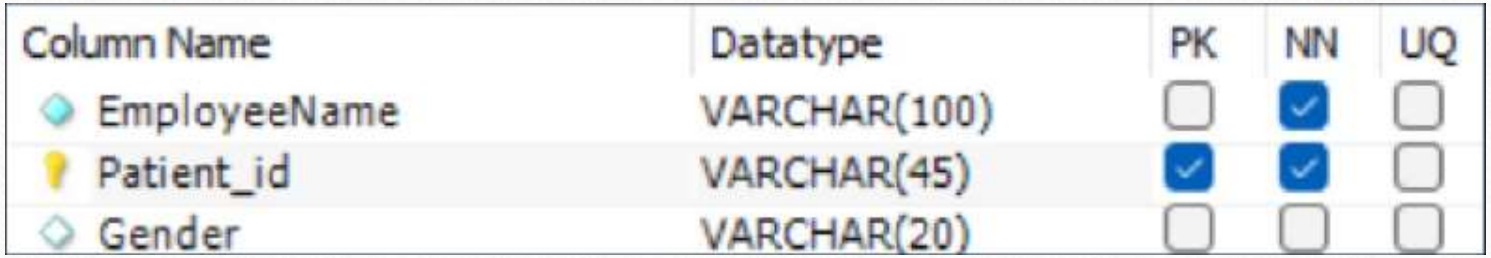


```
SELECT
    *
FROM
    diabetes_patients
WHERE
    hypertension = 1
    except
SELECT
    *
FROM
    diabetes_patients
WHERE
    diabetes = 1;
```

EmployeeName	Patient_id	gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
DENISE SCHMITT	PT129	Male	45	1	0	never	26.47	4	158	0
RAY CRAWFORD	PT155	Female	45	1	0	never	23.05	4.8	130	0
KENNETH SMITH	PT161	Male	44	1	0	current	27.86	6.6	145	0
CHARLES SCOTT	PT215	Female	55	1	0	Ex-smoker	34.2	5.7	140	0
SHANNON SAKOWSKI	PT227	Male	79	1	0	Ex-smoker	28.73	6.6	160	0
MARISA MORET	PT241	Female	80	1	0	Ex-smoker	44.06	6.5	160	0

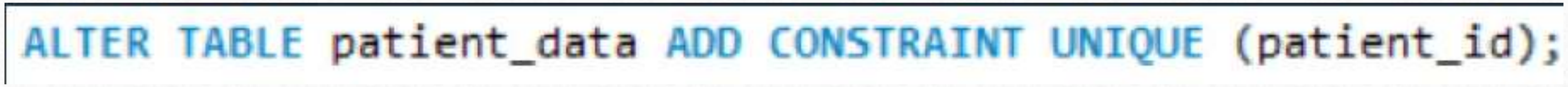
Define a unique constraint on the "patient_id" column to ensure its values are unique.

Before

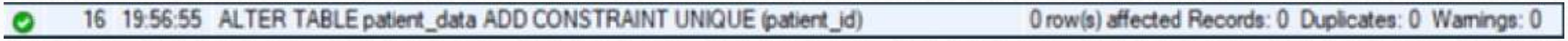


Column Name	Datatype	PK	NN	UQ
EmployeeName	VARCHAR(100)	☐	☑	☐
Patient_id	VARCHAR(45)	☑	☑	☐
Gender	VARCHAR(20)	☐	☐	☐

Query

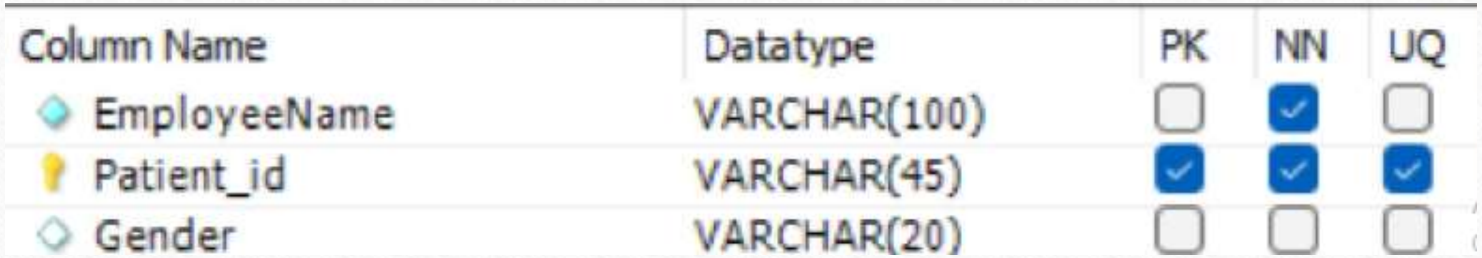


```
ALTER TABLE patient_data ADD CONSTRAINT UNIQUE (patient_id);
```



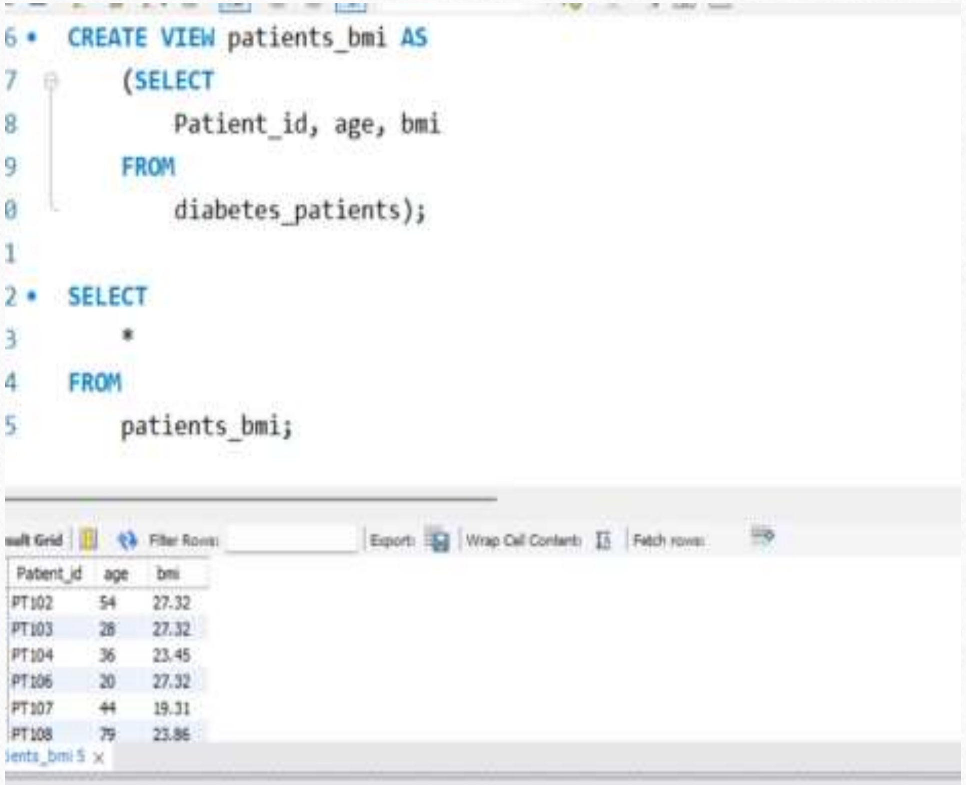
```
16  19:56:55  ALTER TABLE patient_data ADD CONSTRAINT UNIQUE (patient_id)    0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0
```

After



Column Name	Datatype	PK	NN	UQ
EmployeeName	VARCHAR(100)	☐	☑	☐
Patient_id	VARCHAR(45)	☑	☑	☑
Gender	VARCHAR(20)	☐	☐	☐

Create a view that displays the Patient_ids, ages, and BMI of patients.



```sql
CREATE VIEW patients_bmi AS
    (SELECT
        Patient_id, age, bmi
    FROM
        diabetes_patients);

SELECT
    *
FROM
    patients_bmi;
```

Patient_id	age	bmi
PT102	54	27.32
PT103	28	27.32
PT104	36	23.45
PT106	20	27.32
PT107	44	19.31
PT108	79	23.86

Suggest improvements in the database schema to reduce data redundancy and improve data integrity.

- Normalize your database schema to minimize redundancy. This involves breaking down large tables into smaller ones and establishing relationships between them through foreign keys. This ensures that each piece of data is stored only once, reducing redundancy.
- Implement foreign keys to enforce referential integrity between related tables. This ensures that relationships between tables are maintained, preventing orphaned records and ensuring data consistency.
- Define constraints and triggers to enforce business rules and maintain data integrity at the database level. Constraints such as NOT NULL, CHECK, and DEFAULT can help ensure that only valid data is entered into the database.
- Utilize unique constraints on columns where appropriate. This ensures that each value in a particular column is unique, preventing duplicates and improving data integrity.
- Determine the primary key for each entity, which uniquely identifies each record in the table.
- Break down a larger table into smaller tables and establish relationships between them. This reduces redundancy and makes the database more efficient and reliable.

Explain how you can optimize the performance of SQL queries on this dataset.

There are several ways to optimize SQL queries for faster performance, a few are listed below:

1. Minimize the use of wildcard characters.
2. Increase Query Performance with Indexes.
3. Use appropriate data types.
4. Avoid subqueries.
5. Use LIMIT or TOP to limit the number of rows returned.
6. Avoid using SELECT * .
7. Use GROUP BY to group data.
8. Monitor query performance.

THANKYOU